ЛАБОРАТОРНАЯ РАБОТА

НАСТРОЙКА УДАЛЕННОГО АДМИНИСТРИРОВАНИЯ ЗАЩИЩЕННОЙ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ С ИСПОЛЬЗОВАНИЕМ ПРОТОКОЛА SSH

Задание на лабораторную работу

1. На всех компьютерах стенда проверить наличие, при отсутствии установить SSH-сервер, убедится, что он запускается при загрузке ОС.

2. На всех компьютерах стенда проверить наличие, при отсутствии установить SSH-клиент.

3. Зарегистрировать на рабочих станциях и сервере нового пользователя, не имеющего административных полномочий. Имя пользователя формируется из инициалов слушателя, например Сидоров Иван Петрович создает пользователя sip.

4. Сгенерировать для пользователя пару открытый/закрытый ключ SSH на всех компьютерах стенда.

5. Настроить для пользователя двухстороннюю беспарольную аутентификацию по ключу с использованием протокола ssh между рабочими станциями и сервером. Под двухсторонней аутентификацией между рабочей станцией и сервером понимается, что одновременно должны выполняться следующие условия:

– пользователь на сервере может удаленно аутентифицироваться на рабочей станции;

– пользователь на рабочей станции может удаленно аутентифицироваться на сервере.

6. Продемонстрировать настроенную беспарольную аутентификацию для различных взаимодействующих систем путем выполнения из консоли операционной системы действий по:

6.1 удаленному выполнению команды, отображающей имя СВТ, текущее время и дату, имя пользователя, запустившего команду:

- с сервера (Linux) на клиенте (Linux);
- с сервера (Linux) на клиенте (Windows).

6.2 удаленному копированию файла.

- с клиента (Linux) на сервере (Linux);
- с клиента (Windows) на сервере (Linux).

Демонстрацию начать с полной перезагрузки стенда.

УКАЗАТЕЛЬ

Требования к стенду

Для проведения практических занятий и лабораторной работы используется стенд, подготовленный по результатам выполнения предыдущей лабораторной работы с настроенными серверами DHCP и DNS. При невозможности использования результатов предыдущей работы или отсутствия стенда необходимо создать стенд, состоящий из трех виртуальных машин, объединенных сетью. В таком случае настройку серверов DHCP и DNS производить не нужно – необходимо настроить статические IP-адреса сервера и рабочих станций, а также разрешение сетевых имен в IP-адреса с использованием файлов /etc/hosts (для ОС Linux) и c:\windows\system32\drivers\etc\hosts.txt (для ОС Windows).

Краткие теоретические сведения

SSH – сетевой протокол прикладного уровня, позволяющий получать удаленный доступ к компьютеру с большой степенью безопасности соединения, за счет шифрования данных при передаче.

Протокол SSH работает поверх протокола TCP/IPv4 и TCP/IPv6, по умолчанию используется порт 22.

SSH работает по клиент-серверной технологии, то есть имеется SSH-сервер и SSH-клиент.

Сервер SSH – постоянно работает в системе как сервис, принимает запросы от клиентов по протоколу SSH.

Клиент SSH – запускается по команде пользователя, подключается к серверу по протоколу SSH и выполняет определенные действия:

- реализация функции удаленного терминала;
- удаленное выполнение команд;
- защищенное копирование файлов;
- переадресация портов;
- защищенное монтирование удаленной файловой системы.

SSH допускает различные варианты аутентификации, но данной работе будут

использоваться два основных – аутентификация по ключу, а также – по паролю. Аутентификация по паролю является вариантом по умолчанию, доступным сразу после установки SSH. Данный вариант требует действий пользователя по вводу пароля, то есть требует присутствия пользователя в момент выполнения команды. Для построения автоматизированных систем, то есть когда отдельные действия пользователя автоматизируются и выполняются без его участия вариант ввода пароля неприемлем. В таком случае используется вариант аутентификации по ключу, который не требует никаких в момент выполнения команды действий по аутентификации со стороны пользователя или процесса, работающего от имени пользователя. Для второго варианта необходимо произвести предварительные действия по настройке компьютеров. Порядок действий приведен на Рисунке 0.1.

Для возможности использования варианта аутентификации по ключу пользователь USER1 (см. Рисунок 0.1) должен сгенерировать пару открытый-закрытый ключ на компьютере HOST1.

Открытый ключ в дальнейшем будет распространен на компьютеры, где пользователь хочет проходить аутентификацию по ключу. Только пользователь, имеющий закрытый ключ, сможет аутентифицироваться на компьютерах, где установлен открытый ключ.

После генерации ключей, открытый ключ необходимо любым способом скопировать на второй компьютер (HOST2) и произвести там настройку аутентификации по ключу на конкретного пользователя (USER2). Далее пользователь USER1 с компьютера HOST1 с использованием клиента SSH, имея закрытый ключ, может проходить беспарольную аутентификацию по ключу на компьютере HOST2, где установлен сервер SSH с правами пользователя USER2.

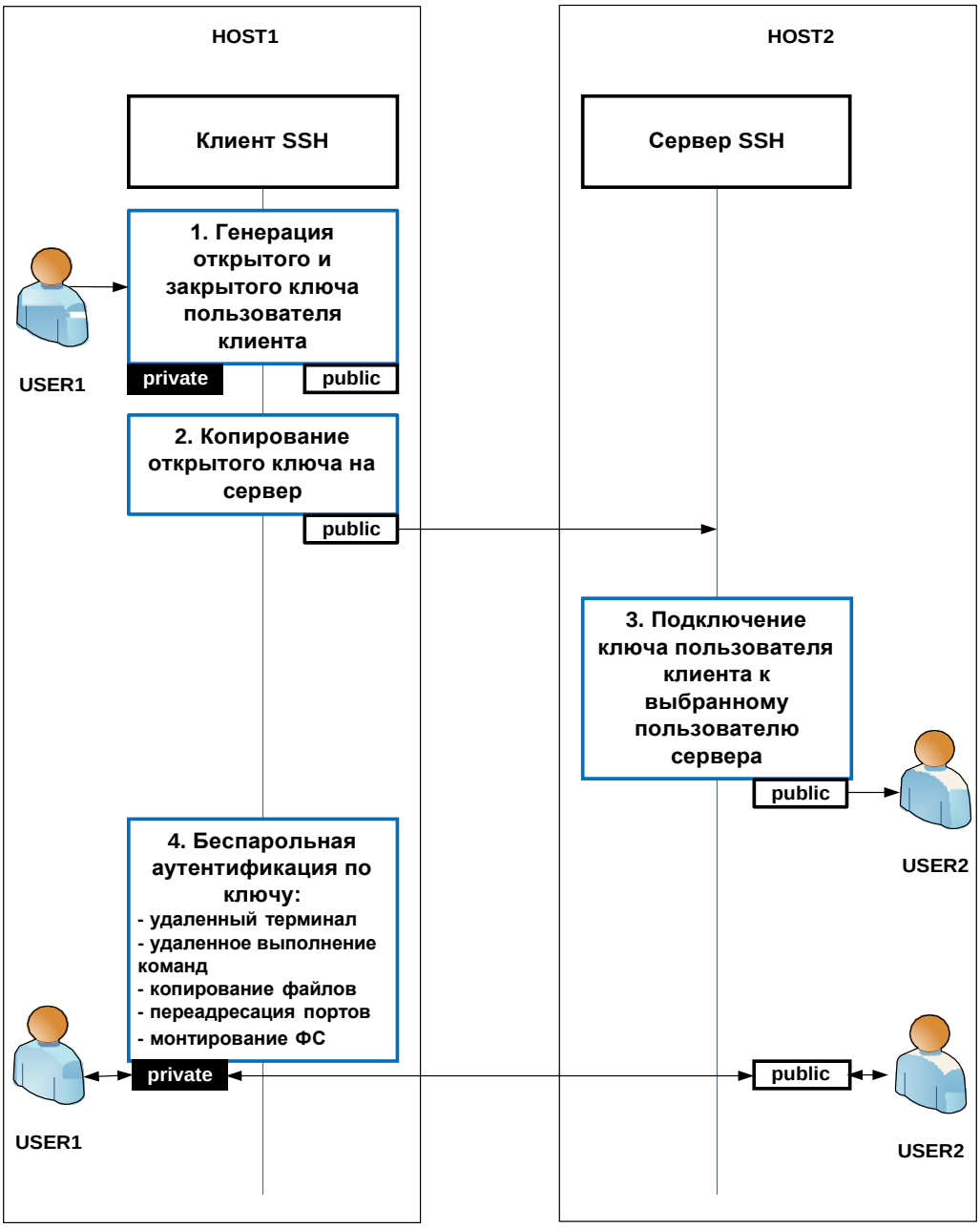

Рисунок 0.1 – Порядок действий для осуществления защищенного подключения по ключу со стороны клиента

Методические рекомендации

Если сервер и клиент SSH не установлены, не обходимо произвести их установку.

Установка сервера SSH в Linux

Проверить наличие сервера можно командой

```
dpkg -s openssh-server
```

При наличии сервера, клиента и общего пакета ssh сообщение будет примерно, таким (возможно отличие в версии пакетов):

```
openssh-server-7.3p1-5-rosa2016.1.x86_64
openssh-clients-7.3p1-5-rosa2016.1.x86_64
openssh-7.3p1-5-rosa2016.1.x86_64
```

Установить сервер без общего пакета и клиента невозможно. При отсутствии необходимых пакетов установить их выполнив команду:

apt install openssh-server

После установки SSH-сервера (sshd) необходимо настроить его автоматический запуск при загрузке компьютера и запустить:

systemctl enable sshd

systemctl start sshd

Если сервер SSH уже установлен, то с большой долей вероятности он запускается автоматически, проверить это можно командой

systemctl status sshd

Проверить работоспособность сервера можно просмотрев список процессов:

ps –ef | grep sshd

Результат будет примерно таким:

```
root 66151 0 13:33 ? /usr/sbin/sshd  -D
```

А также проверив возможность подключения по порту 22:

netstat –ap | grep ssh

Настройки сервера хранятся в файле /etc/ssh/sshd_config

Значения настроек по умолчанию достаточны для выполнения практического задания и **в редактировании файла настроек нет необходимости**, ниже приведены примеры основных параметров.

Используемый порт (22 по умолчанию)

```
Port 22
```

Работа пользователя без пароля:

Запрещена (по умолчанию) / разрешена

```
PermitEmptyPasswords   no / yes
```

Аутентификация суперпользователя:

Запрещена / запрещена без пароля (по умолчанию) / разрешена

```
PermitRootLogin  no / without-password / yes
```

По паролю запрещена, разрешено только удаленное выполнение команд:

```
PermitRootLogin   forced-commands-only
```

Аутентификация по паролю для всех пользователей:

Запрещена / разрешена (по умолчанию)

```
PasswordAuthentication no / yes
```

Аутентификация с использованием открытого ключа:

Разрешена (по умолчанию) / запрещена:

```
PubkeyAuthentication yes / no
```

С учетом того, что существующий по умолчанию файл настроек содержит множество закомментированных настроек, то посмотреть только действующие можно, например, следующей командой, пропускающей строки, содержащие # (знак комментария), и пустые строки:

```
cat /etc/ssh/sshd_config | sed '/#/d;/^$/d'
```

Установка клиента SSH в Linux

Если проведена установка сервера SSH, то клиент установен.

Проверить наличие клиента можно командой:

```
dpkg -s ssh
```

При отсутствии необходимых пакетов установить выполнив команду:

```
apt install ssh
```

Установка сервера SSH в Windows

В Windows 10/11 уже есть встроенный SSH сервер на базе пакета OpenSSH в виде Feature on Demand (FoD). Для установки сервера OpenSSH достаточно выполнить PowerShell команду:

```
Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH.Server*' | Add-WindowsCapability –Online
```

Проверить работоспособность сервера можно путем проверки открытия порта 22 (порт SSH по умолчанию) выполнив команду:

```
netstat -na| find ":22"
```

Результат должен содержать строку

```
Имя Локальный адрес   Внешний адрес Состояние
TCP 0.0.0.0:22        0.0.0.0:0        LISTENING
```

Установка клиента SSH в Windows

В данной работе рассматривается два клиента SSH для ОС Windows – Bitvise SSH Client и Putty. Оба представленных варианта используют как графический интерфейс, так и интерфейс командной строки. Putty имеет более широкий набор консольных приложений, в частности, в нем присутствует программа копирования файлов по интерфейсу SSH, отсутствующая в продукте Bitvise.

Для установки Bitvise SSH Client необходимо запустить файл ***BvSshClient-Inst.exe***. В процессе установки принять условия (См. Рисунок 0.2) и дождаться окончания установки. После окончания установки запустится графический интерфейс клиента SSH, работу в котором более подробно рассмотрим чуть позже (См. Рисунок 0.3).

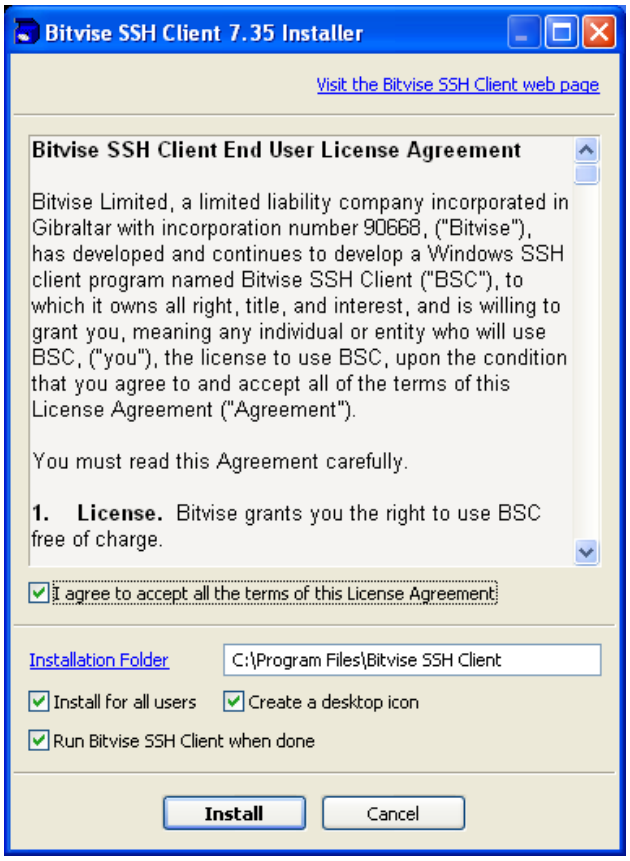

Рисунок 0.2 – Процесс установки Bitvise SSH Client

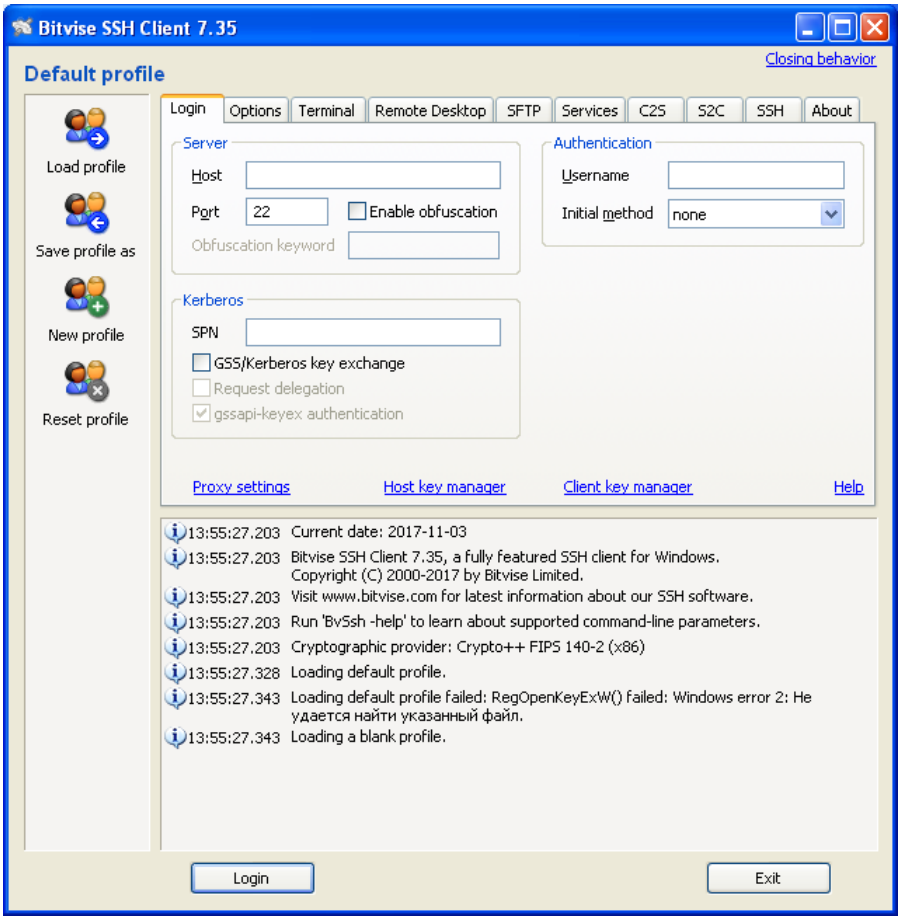

Рисунок 0.3 – Интерфейс клиента SSH от Bitvise

Для работы через интерфейс командной строки в клиенте SSH от Bitvise имеются следующие консольные приложения:

Stermc.exe – консольный клиент SSH (аналог команды ssh в Linux).

Sftpc.exe – консольный клиент FTP.

Sexec.exe – консольный клиент SSH для удаленного выполнения команд (данное действие может быть выполнено командой **stermc.exe** с определенными параметрами).

Для установки второго варианта клиента SSH для ОС Windows – Putty, скопировать в некоторый каталог, из которого будут запускаться команды, либо установить putty-0.70-installer.msi, при этом путь к установленным программам пропишется в системную переменную PATH и команды можно будет вызывать из любой директории (См. Рисунок 0.4). При желании можно распаковать представленный архив putty.zip в любую директорию, при этом необходимо понимать, что переменная PATH не измениться.

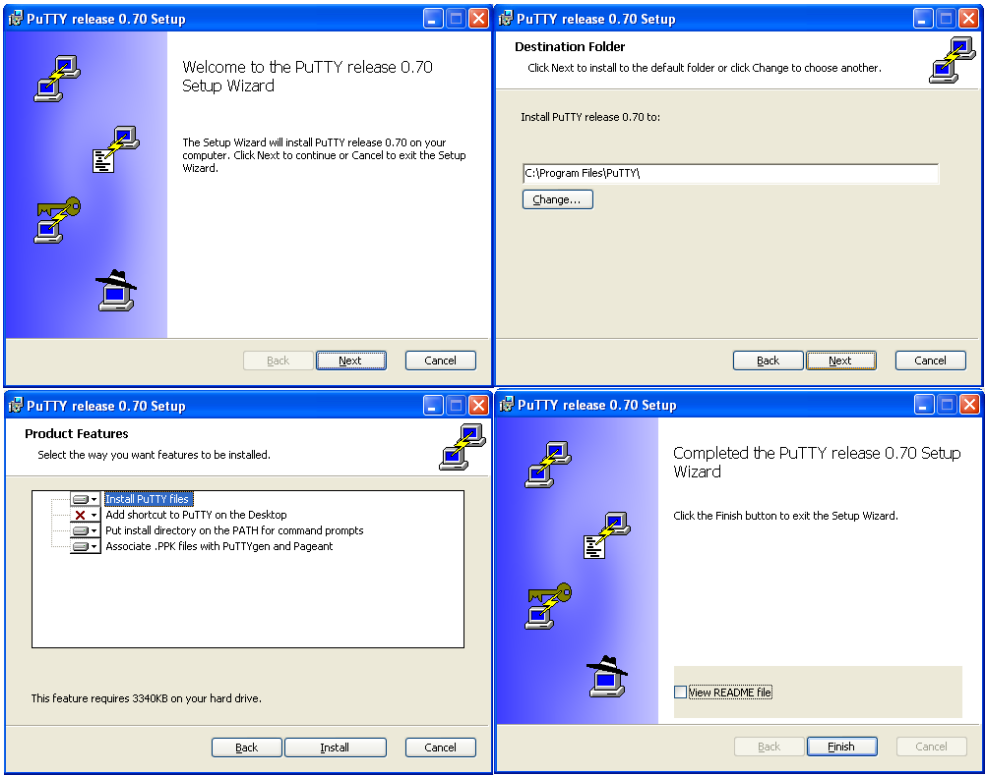

Рисунок 0.4 – Установка клиента SSH Putty

После установки будут доступны следующие программы PUTTY:

Putty.exe – графический клиент SSH.

Plink.exe – консольный клиент SSH (аналог команды ssh в Linux).

Pscp.exe – консольная команда копирования файлов (аналог команды scp в Linux).

Psftp.exe – консольный клиент FTP.

Puttygen.exe – консольная команда генерации и управления ключами (аналог ssh-keygen в Linux).

Генерация ключей в Linux

Для генерации пары открытый-закрытый ключ необходимо выполнить команду:

```
ssh-keygen
```

На все вопросы отвечать по умолчанию, пароль для ключа не вводить, поскольку в противном случае при использовании аутентификации по ключу будет затребован пароль от ключа и, таким образом, потребуется участие пользователя, соответственно, автоматизированное выполнение команд станет невозможным. В нашем случае считается, что доступ к ключу ограничен, и дополнительная его защита не требуется

Результатом выполнения команды является появление в скрытом подкаталоге .ssh домашнего каталога пользователя файлов id_rsa (закрытый ключ) и id_rsa.pub (открытый ключ).

Генерация ключей в Windows

Генерация ключей может производиться как с использованием клиента Bitvise, так и Putty. Клиент Putty может использовать ключи, сгенерированные в клиенте Bitvise, поэтому для упрощения процедуры и сокращения действий будем использовать ключи, сгенерированные в клиенте Bitvise.

Для генерации ключей, используемых клиентом Bitvise на вкладке «Login» (клиента Bitvise) нажать ссылку «Client key manager» и нажать кнопку «Generate New» (См. Рисунок 0.5). Использовать значения по умолчанию, пароль для ключа не вводить, в противном случае, он всегда будет спрашиваться при операции с ключом, то есть требовать действия пользователя, что мешает автоматизации систем.

Запомнить в какой ячейке хранится ключ (См. Рисунок 0.5) по умолчанию

это «global1». По номеру ячейки в дальнейшем будет происходить обращение к ключу.

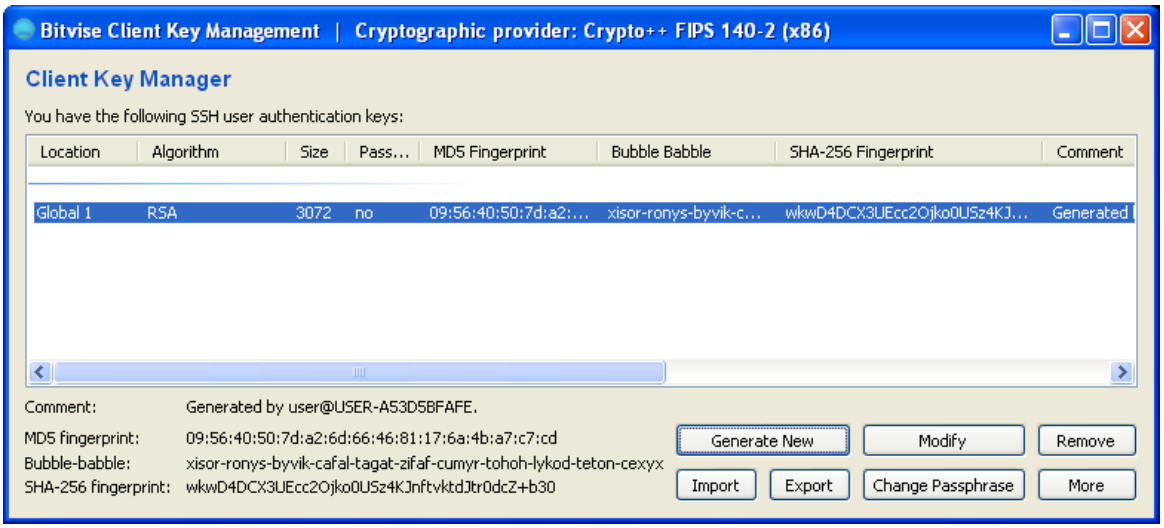

Рисунок 0.5 – Генерация ключа в клиенте Bitvise

Сгенерированная пара ключей будет храниться в Bitvise и может быть использована клиентскими приложениями Bitvise без дополнительных действий. Однако для возможности аутентификации по ключу на других компьютерах необходимо скопировать туда открытый ключ. Для возможности копирования сгенерированного открытого ключа на другие компьютеры необходимо его выгрузить в формате OpenSSH (См. Рисунок 0.6). Данное окно вызывается из «Client key manager» в котором мы генерировали ключ. Сохраните ключ, например, во вновь созданной директории «C:\keys», далее будет подразумеваться, что ключ сохранен в этой директории.

Для возможности использования клиентских программ Putty, не имеющих доступ напрямую к ключам Bitvise необходимо выгрузить закрытый ключ в любую директорию, к которой имеется доступ для дальнейшего использования (См. Рисунок 0.7). Пароли опять не нужно вводить из ранее приведенных соображений.

Рисунок 0.6 – Выгрузка открытого ключа

Рисунок 0.7 – Выгрузка закрытого ключа

Подключение ключей в Linux

Для возможности беспарольной аутентификации по ключу необходимо скопировать открытый ключ пользователя другого компьютера и настроить его отображение в локального пользователя (См. Рисунок 0.1).

Далее приведен пример копирования с использованием команды scp ключа пользователя *user1* с компьютера host*1* в домашний каталог пользователя *user2* компьютера *host2* (команда выполняется на HOST1):

```
scp /home/user1/.ssh/id_rsa.pub
 user2@host2:/home/user2/.ssh/user1.pub
```

Поскольку беспарольная аутентификация по ключу еще не настроена, то в процессе выполнения команды будет запрошен пароль пользователя *user2*. В

данном примере при копировании имя файла открытого ключа пользователя было заменено на *user1.pub* чтобы случайно не перезаписать существующий ключ пользователя *user2* на компьютере *host2*, если он там уже существует.

Далее необходимо добавить открытый ключ пользователя *user1* в файл **authorized_keys** пользователя *user2*, для этого на *host2* в каталоге /home/user2/.ssh/ выполнить

команду:

```
cat user1.pub >> authorized_keys
```

Проверить, что владельцем файла является текущий пользователь и права на файл – 644 командой:

```
ls –l authorized_keys
```

Результат должен быть примерно таким:

```
rw-r—r-- user2 user2 806 authorized_keys
```

При несовпадении изменить владельца и права доступа к файлу на требуемые.

Проверить, что владельцем файла каталога .ssh в домашнем каталоге пользователя является текущий пользователь и права на директорию – 700 командой:

```
ls –la /home/user2/.ssh
```

Результат должен быть примерно таким:

```
drwx---- user2 user2 806 .ssh
```

При несовпадении изменить владельца и права доступа к директории на требуемые.

Важно! Отличие прав доступа и владельца от указанных не позволит работать ssh в режиме доступа по публичному ключу и будет затребован пароль для подключения.

Для проверки успешности настройки беспарольной аутентификации пользователя *user1* компьютера *host1* на компьютере *host2* с правами пользователя *user2* необходимо на компьютере *host1* от имени пользователя *user1* выполнить команду:

```
ssh user2@host2
```

При правильной настройке пароль не будет запрошен и появится удаленный терминал компьютера *host2*.

Для закрытия защищенного подключения ввести команду:

```
exit
```

Подключение ключей в Windows

Запустите обычную (непривилегированную сессию PowerShell) и сгенерируйте пару ключей:

```
ssh-keygen
```

Утилита ssh-keygen создаст каталог **.ssh** в профиле текущего пользователя Windows (%USERPROFILE%\.ssh) и сгенерирует 2 файла:

id_rsa – закрытый ключ

id_rsa .pub – публичный ключ

После того, как ключи созданы, добавите закрытый ключ в службу **SSH Agent**, которая позволяет удобно управлять закрытыми ключами и использовать их для аутентификации.

SSH Agent может хранить закрытые ключи и предоставлять их в контексте безопасности текущего пользователя. Запустите службу ssh-agent и настройте автоматический запуск с помощью PowerShell команд управления службами:

```
Set-service ssh-agent StartupType 'Automatic'
Start-Service ssh-agent
```

Добавьте ваш закрытый ключ в базу ssh-agent:

```
ssh-add "C:\Users\user\.ssh\id_rsa"
```

Скопируйте файл **id_rsa.pub** в каталог **.ssh** профиля пользователя, под которым вы будете подключаться к SSH серверу. Например, в Windows 11 создан пользователь *user1*, значит необходимо скопировать ключ в файл *C:\Users\user1\.ssh\authorized_keys*. Если каталог .ssh в профиле отсутствует, его нужно создать вручную.

По умолчанию в OpenSSH сервере в Windows отключена аутентификация по ключам.

Откройте файл C:\ProgramData\ssh\sshd_config с помощью блокнота, раскоментируйте строку «PubkeyAuthentication yes». Также в конфигурационном файле sshd_config придется отключить режим **StrictModes**. По умолчанию этот режим включен и запрещает аутентификацию по ключам, если закрытый и открытый ключ недостаточно защищены. Раскомментируйте строку #StrictModes yes , измените на StrictModes no.

Сохраните файл и перезапустите службу sshd:

```
Restart-Service sshd
```

Удаленное подключение по протоколу SSH из Linux

При первом подключении в независимости из какой ОС и в какую ОС происходит подключение, возможно будет запрошено разрешение на добавление ключа компьютера в список известных. Необходимо разрешить, в противном случае, такое разрешение будет запрашиваться при каждом подключении.

Для того чтобы, например, пользователь *user1* с компьютера *host1* мог подключиться к компьютеру *host2* с правами пользователя *user2* необходимо на компьютере host1 выполнить от имени пользователя *user1* команду:

```
ssh user2@host2
```

При этом, если беспарольная аутентификация настроена правильно, пароль запрошен не будет и появится консоль компьютера *host2.*

В случае необходимости диагностики подключения необходимо использовать ключ –v, при этом на экран будет выводиться информация об используемых ключах, протоколах шифрования, портах, а также об ошибках:

```
ssh -v user2@host2
```

Удаленное подключение по протоколу SSH из Windows

При использовании графического интерфейса клиента SSH от Bitvise для беспарольной аутентификации необходимо на вкладке «Login» в параметре «Initial method» указать «public key», то есть подключение по открытому ключу и в параметре «Client key» указать ту ячейку, где хранится ключ пользователя (по умолчанию, «Global 1»). После этого введя имя удаленного пользователя и адрес компьютера нажать кнопку «Login». (См. Рисунок 0.8). После подключения откроется окно удаленного терминала компьютера, к которому производи-

лось подключение.

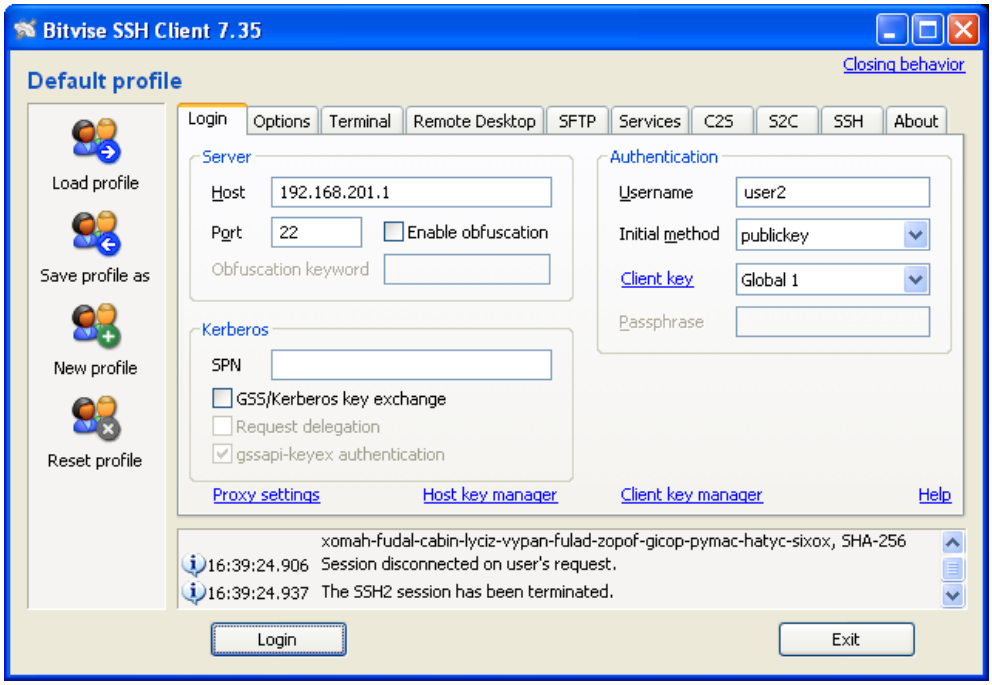

Рисунок 0.8 – Подключение по ключу

При использовании графического интерфейса клиента от Putty (putty.exe) необходимо на вкладке Connection-SSH-Auth (См. Рисунок 0.9) в параметре «Private key for authentication:» указать ранее сгенерированный и выгруженный файл закрытого ключа.

Далее на вкладке Session указать имя или IP-адрес сервера, порт SSH сервера (22 – установлен по умолчанию) и нажать кнопку “Open” (См. Рисунок 0.10). После подключения будет доступен удаленный терминал компьютера.

При использовании клиента SSH от Bitvise для командной строки необходимо выполнить следующую команду, в которой необходимо указать имя пользователя на удаленной машине, адрес или имя удаленной машины и используемый ключ:

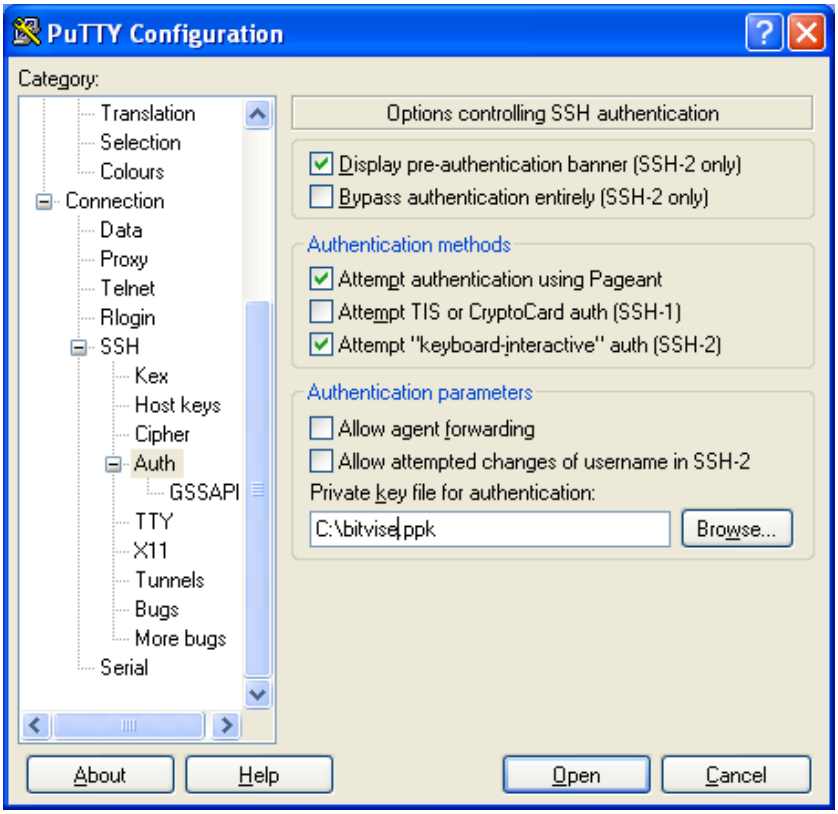

Рисунок 0.9 – Указание закрытого ключа для подключения через Putty

```
stermc user2@192.168.200.1 -pk=global1
```

В данном примере производится подключение к компьютеру с адресом 192.168.200.1 с правами пользователя user2, для беспарольной аутентификации используется ключ, расположенный в ячейке №1.

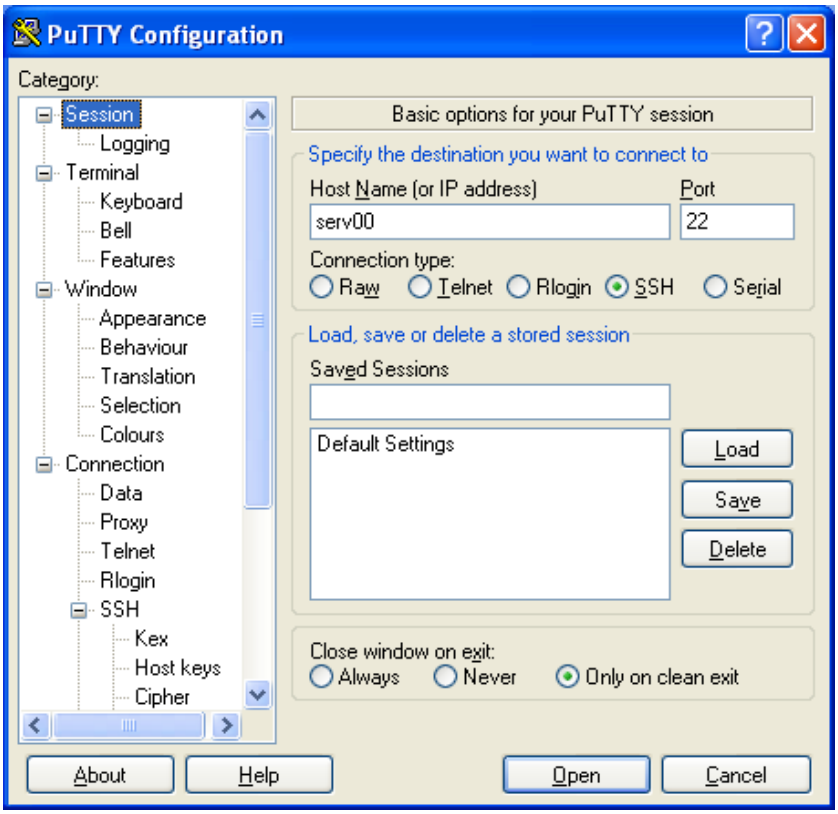

Рисунок 0.10 – Подключение через Putty

При использовании клиента командной строки от Putty необходимо выполнить следующую команду, в которой необходимо указать имя пользователя на удаленной машине, адрес или имя удаленной машины, используемый ключ:

```
plink user2@192.168.200.1 -i c:\bitvise.ppk
```

В данном примере производится подключение к компьютеру с адресом 192.168.200.1 с правами пользователя user2, для беспарольной аутентификации используется закрытый ключ, ранее сгенерированный средствами Bitvise, и сохраненный в файле c:\bitvise.ppk.

Удаленное выполнение команд со стороны клиента SSH из Linux

В Linux для удаленного выполнения команд через SSH используется команда **ssh**, при этом синтаксис не сильно отличается от варианта удаленного подключения.

Для того чтобы, например, пользователь *user1* с компьютера *host1* мог выполнить команду *command* на компьютере *host2* с правами пользователя *user2*, необходимо на компьютере *host1* от имени пользователя *user1*, выполнить команду:

ssh user2@host2 *command*

При этом если команда составная, то есть состоит из нескольких команд через разделитель, необходимо использовать кавычки.

Например, удаленно на Linux можно выполнить команды:

whoami – узнать имя пользователя;

date – узнать текущее время и дату;

hostname – имя компьютера;

pwd – вывести текущий каталог;

uname –a – вывести всю информацию о системе (версию ядра, имя машины и т.д.);

Или их комбинацию через разделитель «;», например, «hostname;whoami».

Например, удаленно на Windows можно выполнить команды:

date /T – узнать текущую дату

time /T – узнать текущее время

hostname – имя компьютера

cd – вывести текущий каталог

ver –вывести версию операционной системы

Или их комбинацию через разделитель «&», например, «hostname&date /T».

Удаленное выполнение команд со стороны клиента SSH из Windows.

При использовании клиента SSH от Bitvise для командной строки необходимо выполнить одну из следующих команд, в которой необходимо указать имя пользователя на удаленной машине, адрес или имя удаленной машины, используемый ключ и команду:

sexec user2@host2 –pk=global1 ls;date

sexec user2@host2 –pk=global1 –cmd=ls;date

stermc user2@host2 –pk=global1 –cmd=ls;date

В данных примерах производится подключение к компьютеру с именем host2 с правами пользователя user2, для беспарольной аутентификации используется ключ расположенный в Bitvise в ячейке №1 и дальнейшее последовательное выполнение команд «ls» и «date».

Использование команды sexec предпочтительнее команды stermc поскольку вторая выводит дополнительную служебную информацию и может быть неправильно интерпретирована при автоматизации действий на стороне клиента.

При использовании клиента от Putty необходимо выполнить следующую команду, в которой необходимо указать имя пользователя на удаленной машине, адрес или имя удаленной машины, используемый ключ и команду:

```
plink user2@host2 –i c:\keys\bitvise.ppk ls;date
```

В данном примере производится подключение к компьютеру с именем host2 с правами пользователя user2, для беспарольной аутентификации используется закрытый ключ, ранее сгенерированный средствами Bitvise, и сохраненный в файле c:\bitvise.ppk и дальнейшее последовательное выполнение команд «ls» и «date».

Удаленное копирование файлов со стороны клиента SSH из Linux

Удаленное копирование осуществляется с использованием команды **scp**.

Для того чтобы, например, пользователь *user1* с компьютера *host1* мог скопировать файл /tmp/file1 на компьютер *host2* с правами пользователя *user2* и сохранить под именем /tmp/file2 необходимо на *host1* выполнить команду от имени пользователя *user1*:

```
scp /tmp/file1 user2@host2:/tmp/file2
```

Примечание: Находясь в ОС Linux при копировании из/в ОС Windows необходимо при использовании полного пути к файлу в ОС Windows использовать кавычки или двойной слеш, так как в Linux одинарный слеш, используемый в Windows как разделитель директории, воспринимается как предшествующий специальному символу:

```
scp /tmp/file user2@host2:”c:\tmp\file”
scp /tmp/file user2@host2:c:\\tmp\\file
```

Удаленное копирование файлов со стороны клиента SSH из Windows

При использовании клиента от Putty необходимо выполнить следующую команду, в которой необходимо указать исходный файл и конечный файл, при этом если какой-то из файлов находится на удаленной машине, то необходимо

указать имя пользователя на удаленной машине, адрес или имя удаленной машины:

pscp –i c:\keys\bitvise.ppk c:\test.txt user2@host2:

В данном примере производится копирование локального файла c:\test.txt на удаленную машину с именем host2 с правами пользователя user2, для беспарольной аутентификации используется закрытый ключ, ранее сгенерированный средствами Bitvise, и сохраненный в файле c:\bitvise.ppk. Поскольку имя конечного файла не указано, то он копируется в домашнюю директорию с тем же именем. В следующем примере производится обратное копирование того же файла в файл c:\test2.txt.

pscp –i c:\keys\bitvise.ppk user2@host2:test.txt c:\test2.txt

Формат команды защищенного копирования SCP (PSCP)

Формат команд защищенного копирования выглядит следующим образом

scp source target, где

source – источник, то есть файл(ы), которые копируются (на локальной или удаленной СВТ).

target – назначение, то есть то куда файл(ы) копируются (на локальной или удаленной СВТ).

Формат *source и target:*

user@host:file , где

user – имя пользователя

host – сетевое имя СВТ

file – наименование файла в любых существующих вариантах (полное имя, короткое имя в локальной директории, множество файлов, выбираемых через *).

Примеры:

Копирование с удаленного компьютера *example.com* файла */etc/hosts* с правами пользователя *fred* на локальный компьютер в файл *c:\temp\example-*

hosts.txt

scp fred@example.com:/etc/hosts c:\temp\ example-hosts.txt

Копирование файла *c:\documents\foo.txt* с локального компьютера на компьютер example.com с правами пользователя *fred* в файл */tmp/foo*

scp c:\documents\foo.txt fred@example.com: /tmp/foo

Копирование всех файлов из каталога *source* с расширением **.c* на удаленном компьютере *example.com* с правами пользователя *fred* в локальный каталог c:\source

scp fred@example.com:source/*.c c:\source

Дополнительные ключи команды доступны путем запуска программы pscp на Windows или man scp на Linux.